

Grunnatriði stýrikerfa

Forritunarverkefni 2- Röðun Ferla

First Come First Serve:

Ég ákvað að besta leið til þessa að skipta á milli mismunandi aðferðana væri einfaldlega bara að copy-pasta Switch-Caseið frá startScheduling inn í processAdded og processFinished til að losa mig við hausverkinn um hvernig á að greina hvaða aðferð frá hvor annari.

Ég bjó líka til klasan ProcessData til að halda við nauðsýnlegu gögnunum sem fara inn í LinkedList Queueið sem ég ætla að nota til þess að raða aðferðirnar.

Að útfæra First Come First Serve var mjög auðvelt að útfæra.

Ég bætti hverju processu í queueið eins og það kom in, gáði ef listin var af stærð 1 til að koma í veg fyrir að eitthvað annað process myndi taka yfir áður en það sem var enþá að keyra kláraði. Þannig að aðeins það allra fyrsta process fær að keyra sitt process í processAdded, fyrir utan það atvik þegar allt í queueinu klárar en það eru enþá fleiri processes sem eiga eftir að jafnvel komast inn í queueið.

Svo í hvert skipti þegar eitthvað process hafði klárað að keyra og kom í processFinished, þá lét ég queueið fjarlægja fyrsta gildið sem var í queueinu, sem er processið sem var að klára að keyra, og ef queueið var ekki tómt átti það að keyra næsta stakið í queueinu.

Round Robin

Ég hannaði Round Robin með því að byrja á því að búa til nýjan þráð sem keyrir þegar það er komið að roundRobin aðferðinni. Svo er ég líka með check í aðferðinni sem drepur þráðinn ef hann skyldi líka enþá vera að keyra ef einhver önnur aðferð er í gangi. ProcessAdded hjá Round Robin er nákvæmlega alveg eins og hjá First-Come-First-Serve, ogeini munurinn á processFinished á First-Come-First-Serve og Round Robin er að ég er ekki með neitt empty check á Queue lista mínum, ég bara tek út efsta stakið í queueinu, þar sem ég geri flest alla aðferðina inn í þræðinum.

Þegar var að keyra Round Robin tók ég eftir að það gerðist oft að process átti nánast ekkert eftir að klára og myndi fara aftur í queueið svo ég bætti við (+ 10) í offset fyrir quantumið til að koma í veg fyrir þetta. Hægt er alveg að taka þetta offset úr og það heldur áfram að keyra rétt eins og að á.

Inn í þræðinum er ég með temp af týpunni ProcessData sem hálpar mér að gá hvaða þráður það var sem var að keyra seinast til að koma í veg fyrir að ég hendi þræði aftast í röðina þegar hann á að keyra næst, vegna þess að hver þráður dettur úr queueinu þegar hann hefur klárað keyrslu. Ég læt þræðin keyra í while lykkju sem er

alltaf true, og það fyrsta sem þráðurinn gerir í hverri lykkju er að láta þráðin fara sofa í quantum tíma. Eftir það gái ég ef queueið mitt er tómt. Ef queueið er ekki tómt þá hef ég if else check þar sem ef fremsta stakið í queueinu er ekki temp þá læt ég þann process keyra og set hann sem temp. Hisvegar ef fyrsta stakið í queueinu er sama og temp þá hendi ég því aftast í röðina, læt sá sem var næstur í röðinni keyra og svo læt ég temp verða sá sem er efstur í queueinu.

Shortest Process Next

Til að auðvelda mér vinnu ákvað ég að nota Comparator og PriorityQueue sem tekur inn ProcessData fyrir Shortest Process Next og ber saman totalServiceTime til þess að raða queueinu í réttri röð. Fyrir utan það þurfti ég líka búa til tvö boolean gildi, firstEnter og FirstExit, til að sjá til um fyrsta stakið í priorityqueueinu þar sem að eina leiðin til að láta PriorityQueue skila gildum í raðaðri röð er að taka þau út úr röðinni. Í processAdded byrja ég á því að bæta gildi inn í queueið og svo gái ég ef að firstEnter er false. Ef það er false þá nota ég switchToProcess á gildið sem var að koma inn. Ég breyti svo firstEnter í true til að koma í veg fyrir að ný gildi breyti yfir á sig þegar þau bætast við og að þetta getur bara gerst þegar priorityQueue er tómt.

Inn í processFinished þá gái ég ef firstExit sé false og ef svo þá breyti ég því í true, og tek út það gildi þar sem ég get ekki leyft næsta process að komast að nema með því að taka það úr queueinu og þar sem fyrsta processið sem er kveikt á er fyrir, þarf að taka það sérstaklega fram að taka það úr.

Svo gái ég ef queueið er tómt, ef svo er þá endurstilli ég firstExit og firstEnter sem false til að leyfa næsta process sem bætist við að byrja ferlið upp á nýtt.

Ef það er ekki tómt þá poll() ég af queueinu og leyfi því process að komast að.

Shortest Remaining Time

Ég gerð svipað og Shortest Process Next í það að ég bjó til Comparator fyrir priorityQueueið og lét það bera saman tíman sem var eftir, þ.e.a.s. (totalServiceTime – executionTime).

Ég þurfti að halda um hvaða process væri sá sem var í gangi án þess að bæta honum inn í queueið þannig að ég bjó til ProcessData sem hét shortestOne og null stilti hana í byrjun:

Svo þegar ég keyri aðferðina inn í processAdded, ef að shortestOne er null þá geri ég nýja processið að shortestOne og kem því í gang

Ef ekki þá þýðir það að það er allavega komin annar process til að keppast fyrir að keyra. Þá uppfæri ég shortestOne miðað við hversu lengi hann er búin að keyra og ber svo hann við þann nýja.

Ef shortestOne er stærri en nýja þá set ég fyrverandi shortestOne í queueið, læt það nýja fljóta verða shortestOne og svo set ég það process í gang.

Ef ekki þá bæti ég bara því nýja í queueið.

Í `processFinished` þá gái ég bara til um hvort `queueið` er tómt. Ef það er þá geri ég `poll()` á `shortestOne` sem setur gildið í `shortestOne` og svo leyfi ég processinu að fá aðgang.

Ef `queueið` er tómt þá þarf ég bara að segja að `shortestOne` er null til að leyfa kerfinu að byrja aftur þegar ný process koma inn.

Highest Response Ratio Next

Hér gerði ég aðferðina næstum nákmælega eins og í Shortest Process Next.

Ég bjó til nýjan `comparator` sem bar á milli `ratio` hjá öllum processum og raðaði þeim í röð.

`processAdded` er nákvæmlega eins og í Shortest Process Next.

Munurinn kemur inn í `processFinished` þar sem ég breyti virknini á um hvort `priorityQueue` er tómt eða ekki.

Ef það er ekki tómt þá fer ég í gegnum allt `priorityQueueið`, `poll()` öllum gildunum og set þau í annað `queue`, `processQueue`. Eftir það þá fer ég í gegnum öll gildin í `processQueueinu` og bæti hverju `ProcessData` aftur inn í `priorityQueue` nema með þeim breytingum að ég hef bætt við þeim færíbreytum sem það `priorityQueueið` þarf til að reikna aftur rétt upp á nýtt hvar hvert `Process` á að vera í röðinni. Svo eftir það set ég næsta process í gang og auðvitað hreinsa ú `processQueueið` svo það eyðileggur ekki fyrir í næstu kall á fallið.

Feedback

\\::TODO:://